

Penerapan *Bee Colony Optimization Algorithm* untuk Penentuan Rute Terpendek (Studi Kasus : Objek Wisata Daerah Istimewa Yogyakarta)

Danuri^{*1}, Widodo Prijodiprodjo²

¹Jurusan Teknik Informatika, Politeknik Negeri Bengkalis, Bengkalis

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: ^{*1}danuri@polbeng.ac.id, ²widodo@ugm.ac.id

Abstrak

Pencarian rute terpendek merupakan suatu permasalahan optimasi yang sering dijadikan studi kasus bagi penelitian. Jarak merupakan faktor yang paling menentukan dalam melakukan penelusuran jalur-jalur yang akan dilalui. Jalur dengan jarak terpendek akan dipilih sebagai jalur pilihan.

Algoritma *bee colony optimization* digunakan dalam penelitian ini untuk menyelesaikan permasalahan pencarian rute terpendek. Terdapat dua proses utama pada saat penelusuran jalur yaitu *forward* dan *backward*. Algoritma *bee colony optimization* bekerja pada proses *forward*. Nilai probabilitas suatu jalur dijadikan dasar pada proses transisi jalur kemudian durasi *waggle dance* dari tiap lebah yang berhasil menemukan posisi tujuan akan dijadikan rute pilihan.

Hasil yang diperoleh dalam penelitian ini adalah algoritma *bee colony optimization* dapat digunakan untuk menemukan rute terpendek. Jumlah lebah yang dilepas sangat mempengaruhi dalam menemukan rute-rute yang bisa dilalui. Semakin banyak jumlah lebah yang dilepas semakin besar peluang ditemukannya rute terpendek.

Kata kunci— Rute Terpendek, Algoritma *Bee Colony Optimization*.

Abstract

The shortest path determination is an optimization problem which often used as a case study for research. Distance is the most defining factor in performing the search paths to be passed. Path with the shortest distance would be chosen as a path selection.

Bee colony optimization algorithm used in this study to complete problems shortest path determination. There are two main process es during search path that is *forward* and *backward*. *Bee colony optimization algorithm* works on the process *forward*. The value probability of a path is base intransition process and the duration of *waggle dance* track of every bee who had found the position of the goal will be a preferred route.

The results obtained in this study is the *bee colony optimization algorithm* can be used to find shortest path. The number of bees are released greatly affects in finding routes that can be passed. The more the number of bees that removed the greater the chances of finding the shortest path.

Keyword— Shortest Path, *Bee Colony Optimization Algorithm*

1. PENDAHULUAN

Rute terpendek merupakan salah satu topik yang menjadi bahasan pada masalah komputasi. Teknik penyelesaian masalah rute terpendek telah menerapkan beberapa metode, seperti algoritma dijkstra, dan algoritma warshall [1]. Metode lain yang digunakan dalam penyelesaian masalah rute terpendek yaitu *branch and bound*, *nearest neighbour*, algoritma genetika, *Ant colony algorithm*, dan *bee colony optimization*.

Masalah rute terpendek yang diselesaikan dengan *bee colony optimization* pernah diteliti dengan kasus *travelling salesman problem*. Penelitian yang dilakukan tersebut dalam membangun tur perjalanan menggunakan konsep pencarian lokal dengan teknik 2-opt. Faktor yang dipertimbangkan pada penelitian tersebut adalah jarak tempuh. Penelitian tersebut diterapkan hanya untuk 1 tujuan [2].

Penelitian pada kasus *travelling salesman problem* juga dilakukan dengan metode penyelesaian *bee colony optimization*, namun dengan menerapkan pencarian lokal yang berbeda. Pencarian lokal yang diteliti menerapkan teknik *frequency-based pruning strategy* (FBPS) dan *fixed-radius near neighbour* (FRNN) 2-opt. Jarak merupakan satu-satunya faktor yang menjadi pertimbangan pada penelitian tersebut. Jumlah posisi tujuan pada penelitian yang telah dilakukan tersebut berjumlah 1 tujuan [3].

Penelitian yang akan dilakukan ini masih menggunakan algoritma *bee colony optimization* dan faktor yang dipertimbangkan adalah jarak tempuh tanpa mempertimbangkan masalah-masalah yang dapat mempengaruhi waktu tempuh seperti kemacetan, lebar jalan, dan *traffic light*. Meskipun demikian ada hal lain yang menjadi titik perbedaan. Pada penelitian sebelumnya pembangunan tur menggunakan teknik pencarian lokal 2-opt dan diteliti kembali menggunakan FBPS dan FRNN 2-opt. Penelitian yang akan dilakukan ini teknik pencarian lokal menggunakan konsep *forward* dan *backward*. Perbedaan lain yang menjadi titik penelitian adalah menerapkan algoritma *bee colony optimization* dengan konsep *exhaustive search* untuk multi tujuan. Penggunaan konsep *exhaustive search* dilakukan untuk mendapatkan solusi optimal secara global dari sisi jarak. Penelitian sebelumnya yang telah dilakukan, banyak posisi atau titik yang menjadi tujuan hanya 1 tujuan.

Algoritma *bee colony optimization* diterapkan pada masalah kunjungan ke objek wisata yang ada di Daerah Istimewa Yogyakarta. Kunjungan yang dilakukan tidak mempertimbangkan masalah jam berkunjung pada objek wisata yang dituju dan tidak ada prioritas kunjungan terhadap objek wisata tertentu. Pencarian rute terpendek diawali dengan mengetahui posisi asal dan tujuan wisata yang akan dikunjungi. Posisi asal dan tujuan wisata yang telah diketahui akan digunakan pada proses pembangunan tur perjalanan dan pencarian rute terpendek dengan algoritma *bee colony optimization*.

2. METODE PENELITIAN

Swarm intelligence merupakan sebuah metode penyelesaian masalah yang memanfaatkan perilaku sekumpulan agen yang saling bekerja sama. Setiap upaya untuk merancang algoritma atau didistribusikan pemecahan perangkat terinspirasi oleh perilaku kolektif dari serangga sosial koloni dan masyarakat hewan lainnya [4].

2.1 Bee Colony Optimization

Lebah merupakan serangga sosial yang sangat terorganisir. Koloni lebah buatan bersama-sama mencari solusi optimal dari masalah yang diberikan. Setiap lebah buatan menghasilkan satu solusi untuk masalah ini. Ada dua fase dalam satu langkah algoritma BCO yaitu fase maju (*forward pass*) dan fase mundur (*backward pass*) [5,6].

Lebah menggunakan aturan transisi dalam membuat keputusan untuk memilih kota yang dikunjungi berikutnya. Probabilitas transisi ($P_{ij,n}$) mengukur kemungkinan perpindahan dari

kota i ke kota j pada transisi n. Probabilitas transisi fungsi jarak dari 2 kota dan *arc fitness* pada jalur yang dilalui. Fungsi ini diformulasikan pada persamaan (1)[2].

$$P_{ij,n} = \frac{[\rho_{ij,n}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}}{\sum_{j \in A_{i,n}} \left([\rho_{ij,n}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}\right)} \quad (1)$$

dimana,

$P_{ij,n}$ = Probabilitas transisi

i = Posisi asal (titik atau kota ke i)

j = Posisi tujuan (titik atau kota ke j) yang bisa ditempuh dari posisi asal

n = Transisi

ρ = *Arc fitness* suatu jalur

d = Jarak

α = Variabel biner yang menonaktifkan pengaruh *arc fitness* dalam model.

β = Parameter yang mengontrol tingkat signifikan jarak.

Pembobotan nilai *arc fitness* (ρ) dilakukan menggunakan persamaan (2)[2].

$$\rho_{ij,n} = \begin{cases} \lambda & , j \in F_{i,n}, |A_{i,n}| > 1 \\ \frac{1-\lambda|A_{i,n} \cap F_{i,n}|}{|A_{i,n} - F_{i,n}|} & , j \notin F_{i,n}, |A_{i,n}| > 1 \\ 1 & , |A_{i,n}| = 1 \end{cases} \quad (2)$$

dimana,

$A_{i,n}$ = Suatu set kota yang bertetangga dengan posisi asal i pada transisi n

$F_{i,n}$ = Satu kota yang merupakan bagian dari $A_{i,n}$ yang dipilih oleh lebah pada transisi n

λ = Nilai *arc fitness* (ρ) untuk jalur yang dipilih oleh lebah

Sekembalinya lebah ke sarang setelah membangun tur lengkap, *waggle dance* akan dilakukan untuk diperlihatkan bagi lebah lain yang ada disarang. Kebijakan yang diterapkan dalam memungkinkan *waggle dance* adalah lebah yang berhasil menemukan sumber makanan yang diperbolehkan untuk menari. Tarian seekor lebah selain memberikan informasi jalan yang lebih pendek juga mengandung informasi durasi waktu. Proses perhitungan *waggle dance* menggunakan persamaan (3) namun dimulai dengan melakukan perhitungan profitabilitas tiap lebah menggunakan persamaan (4), kemudian menghitung profitabilitas koloni lebah dengan persamaan (5)[2].

$$D_i = K \cdot \frac{Pf_i}{Pf_{colony}} \quad (3)$$

$$Pf_i = \frac{1}{L_i} \quad (4)$$

$$Pf_{colony} = \frac{1}{N_{Bee}} \sum_{i=1}^{N_{Bee}} Pf_i \quad (5)$$

dimana,

D_i = durasi *waggle dance* lebah ke i

K = skala faktor *waggle dance*

Pf_i = Profitabilitas lebah ke i

L_i = Panjang jarak yang ditempuh lebah ke i

Pf_{colony} = Profitabilitas koloni lebah

N_{Bee} = Banyaknya lebah yang dilepas dan berhasil menemukan tujuan

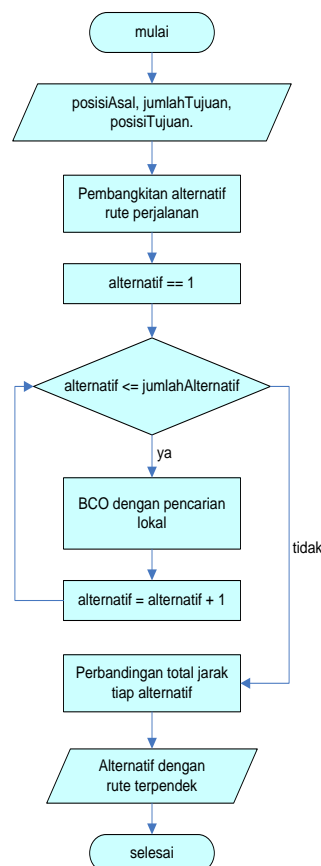
2.2 Perancangan Pencarian Rute Terpendek

Perancangan pencarian rute terpendek menuju objek wisata dengan multi tujuan menggunakan konsep *exhaustive search* dengan menerapkan algoritma *bee colony optimization* ditunjukkan pada Gambar 1. Tahapan pencarian rute terpendek berdasarkan Gambar 1 diawali dengan menerima *inputan* berupa posisi asal (*posisiAsal*), jumlah tujuan wisata (*jumlahTujuan*) beserta objek wisata yang dituju (*posisiTujuan*). Objek-objek wisata yang telah diinputkan akan dilakukan proses pembangkitan alternatif rute perjalanan. Proses pembangkitan alternatif tujuan ini bertujuan untuk mendapatkan kombinasi rute perjalanan dari 1 objek wisata ke objek wisata yang lain sehingga bisa dilakukan penelusuran terhadap semua kemungkinan solusi alternatif rute perjalanan.

Penelusuran pertama yang dilakukan yaitu penelusuran terhadap alternatif rute perjalanan pertama. Penelusuran yang bertujuan untuk mendapatkan rute terpendek pada alternatif rute perjalanan pertama menggunakan algoritma *bee colony optimization* dengan konsep penelusuran untuk pencarian lokalnya adalah *forward* dan *backward*. Hasil penelusuran ini akan didapatkan total jarak keseluruhan yang dibutuhkan dalam menempuh rute perjalanan dari posisi asal menuju semua objek wisata yang diinginkan untuk alternatif rute perjalanan yang pertama.

Penelusuran yang bertujuan untuk mendapatkan rute terpendek dilakukan pada semua alternatif rute perjalanan yang telah dibangkitkan sebelumnya. Apabila ada 6 alternatif rute perjalanan maka akan melakukan sebanyak 6 kali. Penelusuran berakhir setelah sampai pada alternatif rute perjalanan yang terakhir dimana akan diperoleh total jarak tempuh untuk tiap-tiap alternatif rute perjalanan yang ada.

Proses berikutnya adalah membandingkan total jarak tempuh dari semua alternatif rute perjalanan yang ada untuk mendapatkan alternatif rute perjalanan yang memiliki jarak tempuh yang terpendek dari posisi asal menuju semua tujuan wisata yang diinginkan.



Gambar 1 Perancangan pencarian rute terpendek untuk multi tujuan

Perancangan algoritma *bee colony optimization* yang digunakan dalam membangun menemukan rute terpendek dapat dilihat pada Gambar 2.

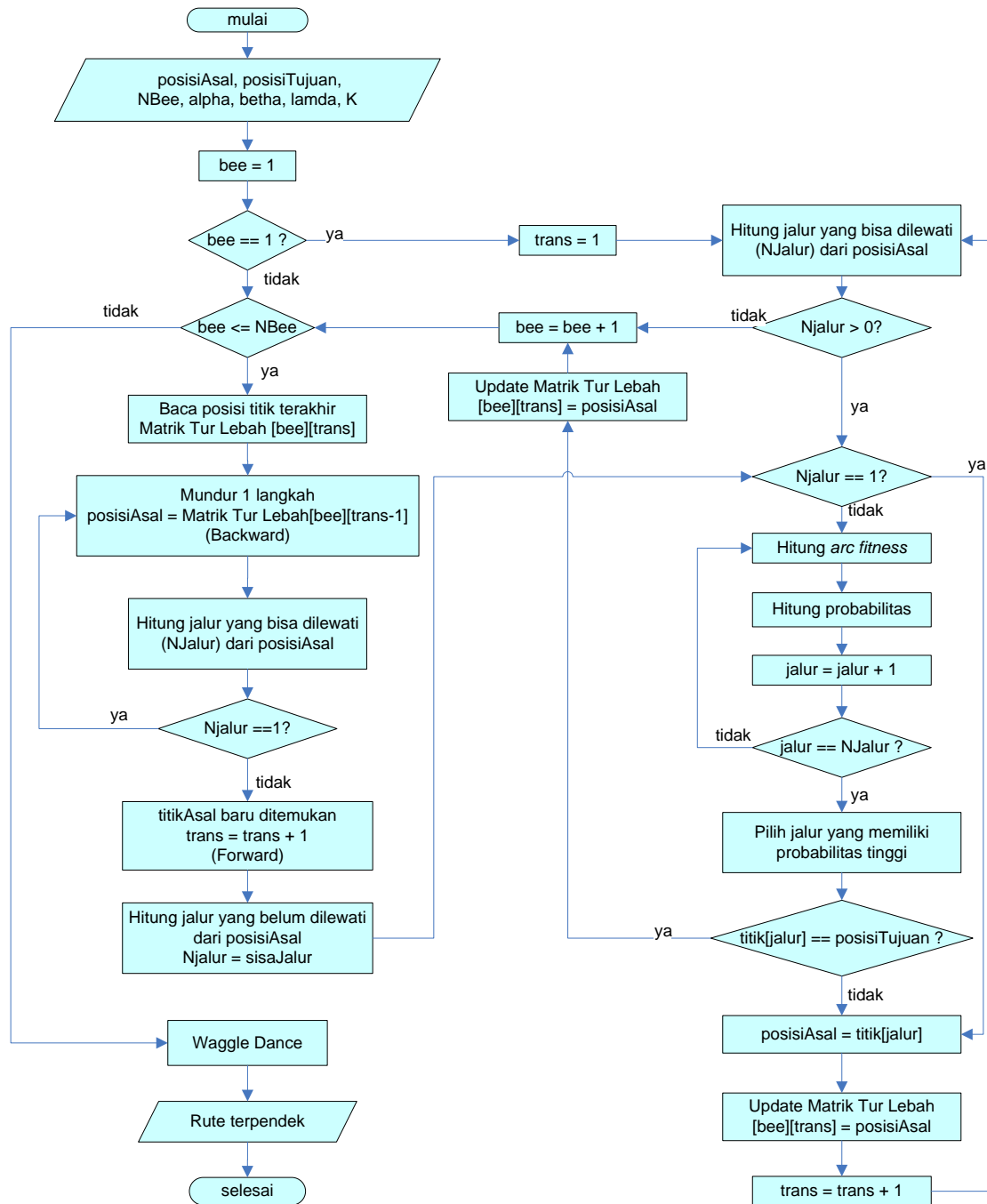
Penelusuran pada tiap alternatif rute perjalanan menggunakan algoritma *bee colony optimization* dengan konsep penelusuran *forward* dan *backward*. Perancangan algoritma *bee colony optimization* dimulai dengan menentukan parameter-parameter yang akan digunakan untuk memaksimumkan siklus lebah. Parameter tersebut adalah posisi asal perjalanan (posisi Asal), posisi tujuan perjalanan (posisi Tujuan), jumlah lebah yang dilepas (NBee), nilai probabilitas untuk jalur yang dipilih λ , variabel biner yang mengubah atau menonaktifkan pengaruh *arc fitness* dalam model (α), parameter yang mengontrol tingkat signifikan jarak (β), dan faktor skala *waggle dance* (K). Posisi asal perjalanan (posisi Asal) merupakan titik yang dijadikan posisi asal perjalanan untuk mencapai posisi tujuan perjalanan (posisi Tujuan). Pada transisi pertama posisi asal diisi oleh posisi asal dimana *user* berada sesuai yang *diinputkan* di sistem, sedangkan pada transisi berikutnya posisi asal diisi oleh titik yang dijadikan titik tujuan pada transisi sebelumnya. Posisi asal perjalanan akan diisi oleh objek wisata yang pertama untuk melanjutkan penelusuran mencapai objek wisata berikutnya setelah mencapai tujuan perjalanan atau objek wisata yang pertama.

Posisi tujuan perjalanan (posisi tujuan) merupakan titik yang dijadikan target penelusuran dari posisi asal. Posisi tujuan diisi oleh objek wisata yang pertama dari urutan yang ada pada alternatif rute perjalanan tertentu, dan akan diisi oleh objek wisata yang kedua apabila proses penelusuran dari posisi asal menuju posisi tujuan (objek wisata pertama) selesai dikerjakan, demikian seterusnya hingga semua objek wisata yang ada di alternatif rute perjalanan tertentu dikunjungi semuanya.

Algoritma *bee colony optimization* bekerja berdasarkan jumlah lebah yang dilepas. Untuk lebah ke-1 penelusuran yang dilakukan adalah penelusuran maju (*forward*). Hal ini berbeda dengan lebah berikutnya. Penelusuran dilakukan dengan cara mengetahui posisi akhir dari lebah sebelumnya ($bee - 1$) kemudian dilakukan penelusuran mundur (*backward*). Setelah menemukan titik yang memiliki pilihan jalur lebih dari 1 dan ada jalur yang belum dilalui oleh lebah-lebah sebelumnya maka titik asal yang baru ditemukan. Titik asal yang baru ini dijadikan posisi awal (posisi Asal) untuk penelusuran maju bagi lebah yang bersangkutan.

Pada saat yang bersamaan terjadi proses penyalinan jalur yang telah dilalui oleh lebah sebelumnya ($bee - 1$). Penyalinan jalur dimulai dari titik asal yang baru sampai ke titik asal bagi lebah sebelumnya ($bee - 1$) atau posisi asal perjalanan. Penelusuran maju (*forward*) dilakukan hingga kriteria penghentian ditemukan.

Waggle dance dilakukan setelah semua lebah selesai melakukan turnya masing-masing. *Waggle dance* dilakukan oleh lebah-lebah yang berhasil menemukan titik tujuan. Bagi lebah-lebah yang tidak menemukan titik tujuan tidak dibenarkan untuk melakukan *waggle dance*. Durasi *waggle dance* yang menunjukkan bahwa rute yang dilalui lebah bersangkutan adalah rute terpendek.

Gambar 2 Diagram alir algoritma *bee colony optimization*

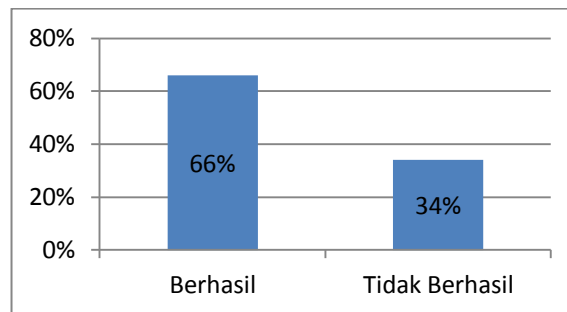
3. HASIL DAN PEMBAHASAN

Pengujian sistem dalam menemukan rute terpendek dilakukan pengujian dengan jumlah tujuan wisata yang berbeda, dan pengujian dengan jumlah lebah yang berbeda.

3.1 Pengujian berdasarkan jumlah tujuan wisata

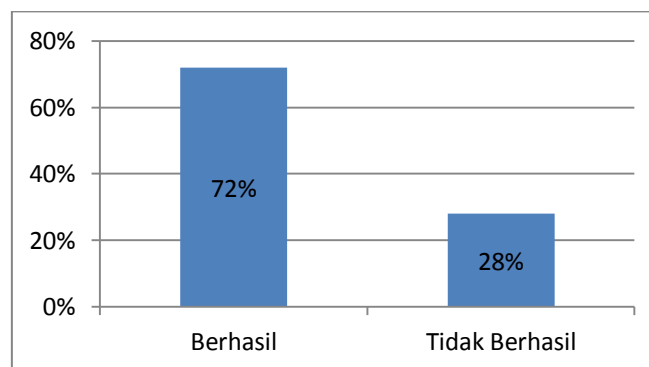
Pengujian berdasarkan jumlah tujuan wisata dilakukan untuk melihat keberhasilan menemukan rute terpendek menuju tujuan wisata. Pengujian ini dibagi menjadi dua yaitu pengujian dengan 1 tujuan wisata dan n tujuan wisata.

Pengujian pertama dilakukan dengan 1 tujuan wisata. Pengujian pertama ini menggunakan parameter yang sama yaitu jumlah lebah (N_{Bee}) = 100, $\alpha = 1$, $\beta = 2$, $\lambda = 0.7$, dan $K = 1$. Secara keseluruhan pengujian dilakukan sebanyak 50 kali. Jumlah tujuan wisata yang diuji sebanyak 10 tujuan wisata dimana setiap tujuan wisata dilakukan pengujian sebanyak 5 kali dengan posisi asal yang berbeda. Pengujian pertama terdapat 33 percobaan yang berhasil dan 17 yang tidak berhasil. Bila dipersentasekan ada 66% percobaan yang berhasil menemukan tujuan wisata dan 34% yang tidak berhasil menemukan tujuan wisata. Persentase tingkat keberhasilan diperlihatkan pada Gambar 3.



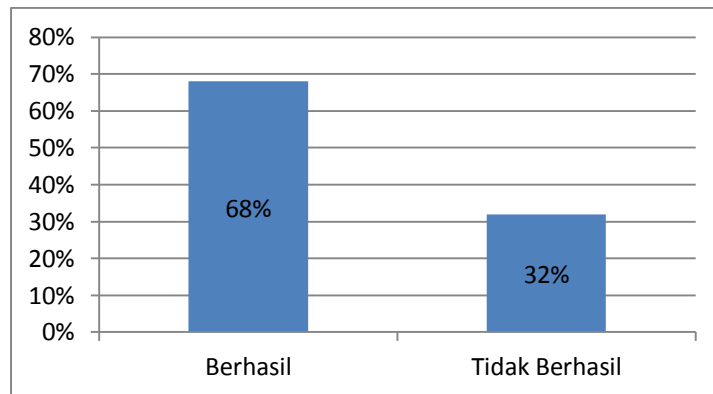
Gambar 3 Grafik perbandingan hasil pengujian dengan 1 tujuan wisata

Pengujian kedua adalah pengujian dengan n tujuan wisata. Pengujian dibagi menjadi dua yaitu dengan 2 tujuan wisata dan 3 tujuan wisata. Pengujian kedua ini menggunakan parameter yang sama yaitu jumlah lebah (N_{Bee}) = 100, $\alpha = 1$, $\beta = 2$, $\lambda = 0.7$, dan $K = 1$. Pengujian dengan 2 tujuan wisata akan dilakukan sebanyak 25 kali. Jumlah kombinasi tujuan wisata yang diuji sebanyak 5 kombinasi dengan 5 posisi asal yang berbeda untuk setiap kombinasinya. Pengujian kedua ini diperoleh hasil yaitu dari 25 percobaan terdapat 18 percobaan yang berhasil dan 7 yang tidak berhasil. Bila dipersentasekan ada 72% percobaan yang berhasil menemukan tujuan wisata dan 28% yang tidak berhasil menemukan tujuan wisata. Tingkat keberhasilan pengujian dengan 2 tujuan wisata dapat dilihat dari grafik pada Gambar 4.



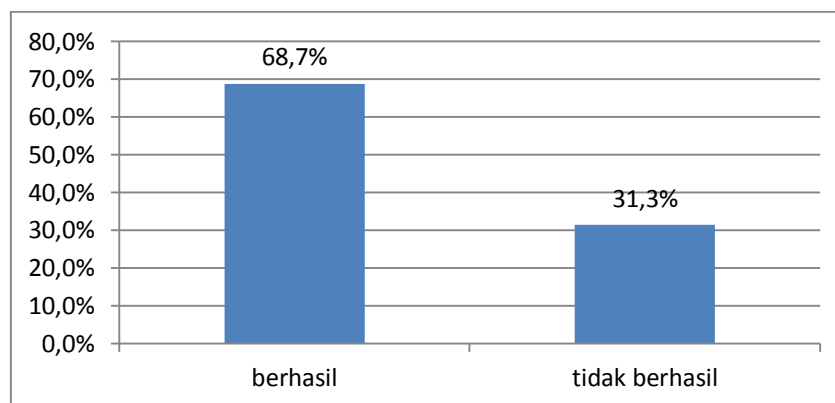
Gambar 4 Grafik perbandingan hasil pengujian dengan 2 tujuan wisata

Pengujian ketiga dilakukan dengan 3 tujuan wisata akan dilakukan sebanyak 25 kali. Jumlah kombinasi tujuan wisata yang diuji sebanyak 5 kombinasi dengan 5 posisi asal yang berbeda untuk setiap kombinasinya. Hasil pengujian ketiga terdapat 17 percobaan yang berhasil dan 8 yang tidak berhasil dari 25 percobaan yang dilakukan. Bila dipersentasekan ada 68% percobaan yang berhasil menemukan tujuan wisata dan 16% yang tidak berhasil menemukan tujuan wisata. Tingkat keberhasilan pengujian dengan 2 tujuan wisata dapat dilihat dari grafik pada Gambar 5.



Gambar 5 Grafik perbandingan hasil pengujian dengan 3 tujuan wisata

Dari 3 pengujian diatas persentase keberhasilan menemukan tujuan wisata rata-rata 68,7%. Pengujian pertama persentase keberhasilan sebesar 66%, pengujian kedua dengan 2 tujuan wisata sebesar 72% dan pengujian dengan 3 tujuan wisata sebesar 68%.Tingkat keberhasilan rata-rata menemukan tujuan wisata dapat dilihat pada Gambar 6.



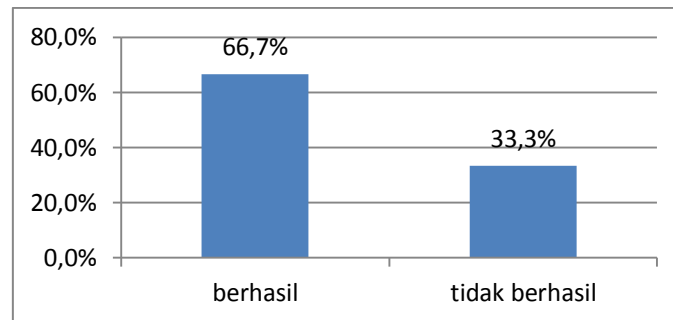
Gambar 6 Grafik tingkat keberhasilan rata-rata

3.2 Pengujian berdasarkan jumlah Lebah

Pengujian berdasarkan jumlah lebah dilakukan untuk melihat pengaruh jumlah lebah yang dilepas dalam menemukan rute terpendek menuju tujuan wisata.

Pengujian pertama menggunakan parameter yang sama yaitu $\alpha = 1$, $\beta = 2$, $\lambda = 0.7$, dan $K = 1$. Pengujian pertama dilakukan dengan 1 ekor lebah. Pengujian berikutnya dengan 10, 50, 100, dan 300 ekor lebah. Secara keseluruhan pengujian dilakukan sebanyak 25 kali dengan 1 tujuan wisata sebanyak 10 kali dan 2 tujuan wisata sebanyak 10 kali dan 3 tujuan wisata 5 kali.

Tujuan wisata yang berjumlah 1 memiliki tingkat keberhasilan sebesar 80%, hal ini ditandai dengan 8 percobaan yang berhasil dari 10 percobaan yang dilakukan. Tujuan wisata yang berjumlah 2 memiliki tingkat keberhasilan sebesar 80%, hal ini ditandai dengan 8 percobaan yang berhasil dari 10 percobaan yang dilakukan. Tujuan wisata yang berjumlah 3 memiliki tingkat keberhasilan sebesar 40%, hal ini ditandai dengan 2 percobaan yang berhasil dari 5 percobaan yang dilakukan. Rata-rata keberhasilan sebesar 66,7%. Tingkat keberhasilan rata-rata menemukan tujuan wisata dibawah pengaruh jumlah lebah dapat dilihat pada Gambar 7.



Gambar 7 Grafik tingkat keberhasilan rata-rata dibawah pengaruh jumlah lebah

3.3 Pengujian berdasarkan parameter

Pengujian parameter α dilakukan mengetahui pengaruh parameter tersebut dalam penentuan rute terpendek. Pengujian parameter dilakukan dengan 2 nilai yaitu 1 dan 0. Nilai parameter lain pada pengujian parameter α yaitu NBee = 50, $\beta = 2$, $\lambda = 0.7$, dan $K = 1$. Pengujian dilakukan sebanyak 20 kali percobaan. Hasil dari pengujian ini yaitu parameter α tidak berpengaruh terhadap penentuan rute terpendek. Berdasarkan teori sebelumnya parameter α merupakan variabel biner yang berfungsi menonaktifkan pengaruh *arc fitness* dari suatu jalur pada proses perhitungan probabilitas untuk transisi.

Pengujian parameter β dilakukan mengetahui pengaruh parameter tersebut dalam penentuan rute terpendek. Nilai yang digunakan untuk pengujian parameter β yaitu 0.1, 0.5, 1, dan 2. Nilai parameter lain pada pengujian parameter β yaitu NBee = 50, $\alpha = 1$, $\lambda = 0.7$, dan $K = 1$. Pengujian dilakukan sebanyak 20 kali percobaan. Parameter β tidak berhasil menemukan rute terpendek untuk nilai dibawah 1, hal ini ditandai dengan percobaan dengan pemberian nilai 0,1 dan 0,5 hanya 1 kali percobaan yang berhasil. Nilai lebih besar sama dengan 1 parameter β bisa menentukan rute terpendek.

Pengujian parameter λ dilakukan mengetahui pengaruh parameter tersebut dalam penentuan rute terpendek. Nilai yang digunakan untuk pengujian parameter λ yaitu 0.1, 0.5, 0.7, dan 1. Nilai parameter lain pada pengujian parameter β yaitu NBee = 50, $\alpha = 1$, $\beta = 2$, dan $K = 1$. Pengujian dilakukan sebanyak 20 kali percobaan. Parameter λ cenderung tidak berhasil menemukan rute terpendek untuk nilai = 1.

Pengujian parameter K dilakukan untuk mengetahui pengaruh parameter tersebut dalam penentuan rute terpendek. Nilai yang digunakan untuk pengujian parameter K yaitu 0.5, 1, dan 2. Nilai parameter lain pada pengujian parameter K yaitu NBee = 50, $\alpha = 1$, $\beta = 2$, dan $\lambda = 0.7$. Pengujian dilakukan sebanyak 18 kali percobaan, parameter K tidak memberikan pengaruh yang cukup signifikan untuk menemukan rute dengan jarak tempuh yang pendek.

3.4 Pengujian dengan perbandingan google map

Pengujian ini dilakukan untuk membanding hasil pencarian rute terpendek berdasarkan sistem yang dibangun dengan aplikasi yang telah ada yaitu *google map*. Nilai parameter yang digunakan pada pengujian ini yaitu NBee = 50, $\alpha = 1$, $\beta = 2$, $\lambda = 0.7$ dan $K = 1$. Pengujian dilakukan sebanyak 10 kali percobaan untuk jumlah tujuan wisata = 1. Hasil dari 10 percobaan dapat diketahui performa *google map* lebih unggul dibanding sistem. Sistem menemukan tujuan wisata dengan jarak yang mendekati penelusuran *google map* sebanyak 4 percobaan. 6 percobaan yang lain sistem berhasil menemukan tujuan wisata tetapi ditinjau dari sisi jarak lebih unggul dengan *google map*.

Pengujian berikutnya dilakukan dengan 2 tujuan wisata. Performa aplikasi *google map* lebih unggul dibanding sistem. Meskipun demikian, sistem memiliki kemampuan memilih tujuan wisata yang terdekat apabila ada 2 atau lebih tujuan wisata yang diinputkan sementara aplikasi *google map* mengerjakan secara berurutan.

3.5 Analisa hasil pengujian

Berdasarkan pengujian untuk mengetahui performa algoritma lebah dalam pencarian rute terpendek didapat hasil sebagai berikut :

1. Pengujian berdasarkan jumlah tujuan wisata didapat hasil berupa algoritma *bee colony optimization* bisa diterapkan untuk menemukan posisi tujuan. Lebah akan mencari tujuan wisata secara keseluruhan dimulai dari posisi asal menuju tujuan wisata yang *diinputkan*
2. Pengujian berdasarkan jumlah lebah didapat hasil berupa peluang menemukan rute terpendek akan lebih besar apabila jumlah lebah yang dilepas semakin banyak. Lebah-lebah yang dilepas tidak semuanya berhasil mencapai posisi tujuan. Lebah yang berhasil mencapai posisi tujuan dipilih berdasarkan *waggle dance* yang terlama. Pengujian dengan jumlah lebah yang kecil cenderung menghasilkan rute terpendek yang kurang optimal. Semakin banyak lebah yang dilepas akan memperbesar peluang ditemukannya rute terpendek dari posisi awal ke posisi tujuan. Hal ini disebabkan sistem akan menelusuri sebanyak mungkin rute yang bisa dilalui sesuai dengan jumlah lebah yang dilepas. Jumlah lebah juga mempengaruhi waktu proses sistem. Semakin banyak jumlah lebah waktu proses akan semakin bertambah. Jumlah lebah yang ideal untuk dilepas adalah antara 10 sampai 100 lebah.
3. Pengujian terhadap parameter didapat hasil berupa lebah-lebah tetap menemukan posisi tujuan. Perubahan parameter mempengaruhi proses perhitungan pada saat penentuan jalur. Parameter α dan β digunakan sebagai pangkat dari formula matematika untuk perhitungan probabilitas transisi. Parameter λ digunakan untuk menentukan besarnya peluang jalur yang dipilih. *Arc fitness* ditentukan parameter α . Parameter α bernilai 0 akan menonaktifkan pengaruh nilai $\rho_{ij,n}$ (*arc fitness* atau kebugaran busur) sehingga nilai yang ditetapkan pada λ tidak berpengaruh. Parameter α bernilai 1 akan mengaktifkan pengaruh nilai $\rho_{ij,n}$ (*arc fitness* atau kebugaran busur) sehingga nilai yang ditetapkan pada λ berpengaruh. Pengaruh signifikan jarak ditentukan parameter β . Nilai β yang optimal bernilai = 2 untuk dapat mengontrol pengaruh signifikan jarak. Nilai λ optimal pada nilai besar dari 0,5 sampai kecil dari 1. Nilai λ yang kecil atau sama dengan 0,5 akan memberikan kemungkinan dominasi antara jalur yang dipilih dengan jalur yang tidak dipilih memiliki nilai yang sama atau bahkan lebih kecil sehingga bisa menyebabkan jalur yang memiliki jarak terpendek tidak terpilih. Durasi tarian dipengaruhi oleh skala faktor K. Nilai K yang optimal adalah 1. Perubahan parameter tidak terlalu memberikan pengaruh yang signifikan.
4. Lamanya proses sistem sangat dipengaruhi oleh banyaknya tujuan wisata dan jumlah lebah yang dilepas. Sistem akan memproses secara bergantian terhadap tujuan-tujuan wisata yang *diinputkan* untuk menemukan rute terpendek. Setiap tujuan wisata akan dikunjungi oleh sebanyak lebah yang dilepas, kemudian dari sekian banyak lebah yang dilepas akan diseleksi lebah mana saja yang berhasil menemukan posisi tujuan. Lebah yang berhasil ini akan melakukan *waggle dance* dan akan diketahui lebah mana yang memiliki durasi tarian yang lama. Lebah yang memiliki durasi tarian yang lama akan dipilih jalur yang dilaluinya sebagai rute terpendek untuk tujuan wisata yang dimaksud. Lebah-lebah yang menemukan rute terpendek untuk setiap tujuan wisatanya kembali akan melakukan *waggle dance* untuk menemukan mana yang lebih terpendek lagi. Hasil dari *waggle dance* ini akan diketahui tujuan wisata mana yang lebih terpendek diantara tujuan wisata yang *diinputkan*.
5. Pencarian rute terpendek menggunakan konsep penelusuran multi tujuan memperbesar peluang ditemukannya rute perjalanan yang lebih pendek sehingga didapat solusi secara *global*. Jumlah tujuan wisata yang *diinput* mempengaruhi banyaknya alternatif rute perjalanan yang bisa ditempuh dari posisi asal menuju keseluruhan tujuan wisata yang *diinputkan*. Penelusuran multi tujuan mempengaruhi waktu proses komputasi. Semakin

banyak tujuan wisata yang diinputkan semakin besar waktu proses komputasi yang dibutuhkan.

Penerapan algoritma *bee colony optimization* untuk mencari rute terpendek secara umum bisa diterapkan. Faktor yang paling mempengaruhi menemukan rute terpendek adalah jumlah lebah yang dilepas untuk menemukan rute terpendek. Sistem yang dibangun ini menelusuri jalur yang mungkin sesuai dengan lebah yang dilepas. Semakin banyak lebah yang dilepas semakin besar peluang menemukan rute yang terpendek.

4. KESIMPULAN

Berdasarkan dari hasil penelitian dan pembahasan yang dilakukan maka diperoleh kesimpulan sebagai berikut:

1. Pencarian rute terpendek menggunakan konsep penelusuran multi tujuan memperbesar peluang ditemukannya rute perjalanan yang lebih pendek sehingga didapat solusi secara *global*.
2. Jumlah lebah yang dilepas mempengaruhi peluang untuk menemukan rute terpendek. Semakin banyak lebah yang dilepas akan memperbesar peluang ditemukannya rute terpendek dari posisi awal ke posisi tujuan.
3. Paramater tidak memberikan pengaruh yang signifikan untuk menemukan rute terpendek

5. SARAN

Rencana pengembangan sistem ke depan untuk meningkatkan performa dan akurasi dari sistem dalam menerapkan algoritma *bee colony optimization* maka saran-saran yang dapat diberikan adalah sebagai berikut:

1. Penelitian dapat dikembangkan dengan menggunakan teknik pencarian yang lain supaya penelusuran lebih optimal. Pada sistem ini penelusuran dilakukan sebanyak mungkin sesuai dengan jumlah alternatif rute perjalanan dan jumlah lebah yang dilepas. Hal ini mengakibatkan lamanya waktu proses dalam menemukan rute terpendek.
2. Penelitian dapat dikembangkan dengan mempertimbangkan faktor-faktor yang dapat mempengaruhi waktu tempuh seperti kemacetan, lebar jalan, dan *traffic light*. Faktor lain yang bisa dipertimbangkan adalah prioritas kunjungan terhadap suatu objek wisata tertentu dan jam berkunjung ke suatu objek wisata.
3. Perlu dilakukan penelitian terhadap kasus yang berbeda untuk mengetahui performa algoritma *bee colony optimization*.

DAFTAR PUSTAKA

- [1] Siang, J. J., 2009, *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*, Andi, Yogyakarta .
- [2] Wong,L., Low, M.Y.H., and Chong, C.S., 2008,Bee Colony Optimization with Local Search for Traveling Salesman Problem, *Proceeding of 6th IEEE International Conference on Industrial Informatics (INDIN 2008)*,Daejeon – Korea, 13 - 16 Juli 2008, 1019 – 1025.
- [3] Wong,L., Low, M.Y.H., and Chong, C.S., 2009, An Efficient Bee Colony Optimization Algorithm for Traveling Salesman Problem using Frequency based Pruning,*Proceedings of 7th IEEE International Conference on Industrial Informatics (INDIN 2009)*, Wales – United Kingdom, 23-26 Juni 2009, 775 – 782.
- [4] Bonabeau, E., Dorigo, M., dan Theraulaz, G., 1999, *Swarm Intelligence from Natural to Artificial Systems*, Oxford University Press, New York.
- [5] Teodorovic, D., 2009, Bee Colony Optimization (BCO),
<http://www.sf.bg.ac.rs/downloads/katedre/oi/1.BCO-Book-Chapter.pdf>, diakses 03 Oktober 2011.
- [6] Kaur, A., Goyal, S., 2011, A Survey on the Applications of Bee Colony Optimization Techniques, *International Journal on Computer Science and Engineering(IJCSE)*, India, 8 Agustus 2011, vol. 3, 3037 – 3046, ISSN : 0975 – 3397.